#20

Docket No.: MS124236.1
Serial No.: 09/364,527

Patent
Docket No. MS124236.1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application:	**EXPEDITED PROCEDURE** **37 C.F.R. 1.116**
Eric HORVITZ	
Serial No.: 09/364,527	Group: 2176
Filed: October 26, 2001	Examiner: R. Singh Confirmation No.: 9577
Title: METHOD FOR AUTOMATICALLY ASSIGNING PRIORITIES TO DOCUMENTS AND MESSAGES	

DECLARATION SUBMITTED PURSUANT TO 37 C.F.R. § 1.132

Attn: Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

[0001] My name is John Platt. I am presently employed as a Senior Researcher for Microsoft Corporation, the assignee of the above-identified patent application. Attached is a current copy of my curriculum vitae.

[0002] In preparation for making this declaration, I have reviewed the following documents:

- Patent Application Serial No. 09/364,527 as filed with the U.S. Patent and Trademark Office on October 26, 2001.
- Amendment filed May 7, 2003 in Patent Application Serial No. 09/364,527.

- Office Action mailed July 18, 2003 in Patent Application Serial No. 09/364,527.
- U.S. Patent No. 6,408,277 to Nelken

[0003] As regards Patent Application Serial No. 09/364,527 (the application), I understand the Examiner has taken the position that claim 36 is indefinite. I understand that the Examiner has asserted that it is unclear what the implicit training module of claim 36 is configured to do. I disagree for the following reasons. I have read claim 36 and have read the application and find that claim 36 is clear to me in view of the application and my experience, and would be clear to one working in the field. In particular, I understand that the implicit training module is "configured to continually watch text selected by a user while working," as set out in lines 5 - 6 of claim 36. The two features of "having an assigned priority" and "comprising new training messages to the text classifier" are each directed to describing the "selected text," i.e., the "text selected by a user while working." A comma is inserted in the claim after "a user while working" and in front of "the selected text" to specify that the "assigned priority" and "new training messages" apply to the "text selected by a user." This meaning is consistent both with the explanation in the application at page 10, lines 5 - 6; and page 15, lines 20 – 29 and with my experience. Therefore, to answer the question presented in the Office Action, the implicit training module is configured to continually watch text selected by a user; and the selected text has an assigned priority and comprises new training messages to the text classifier.

[0004] I also understand that the Examiner has asserted in the Office Action that the derivation of the term, "assigned priority," in line 6 of claim 36 is unclear. I have read the application and find that this term is explained at page 15, lines 20 - 29; page 16, lines 20 - 25; and page 26, lines 11 - 13. Text which is to have a priority assigned is received by embodiments of the system (see page 26, line 11). A priority for each text is generated by the text classifier (page 26, line 13 and page 16, lines 20 - 21), is assigned to the text (page 16, line 21 and page 26, line 11), and is output to storage (page 26, line 16). The user selects the prioritized text while the system watches, and the text classifier is periodically updated, with the selected prioritized text constituting new training messages to the text classifier.

[0005] I have been asked to compare this patent application with U.S. Patent No. 6,408,277 (the Nelken patent) and comment on whether there is a distinction between two particular features of the Nelken patent and certain features of the application. I have compared the "task queue" of Nelken with the "storage media" of the application, and have also compared the "task" of Nelken with the "text" of the application.

[0006] As to the terms "task queue" and "storage media," "queue" is a well known term in the art of computer systems for describing a data structure for storing information in sorted order. See, for example, Webster's Computer & Internet Dictionary 460 (Random House, 3d ed. 1999); Dictionary of Electronics and Computer Technology 429 (McGraw-Hill, 3d ed. 1984). In the context of the Nelken patent and the application, both of these terms are acting in an equivalent manner. Software algorithms for ordering objects can be designed to write data equally to queues and storage media, read data from the same, and process data residing on such devices and in such structures. Sorted data can be kept in either queues or storage media. The software priority capabilities of Nelken are not provided by the sorting behavior of its queues. Instead, it is the software priority module 314 of the decision engine 212 (see Figs. 2 and 3) of Nelken that determines the priority of the tasks for storage in the task queue (see Nelken at Col. 3, lines 43 - 47; Col. 4, lines 39 - 40; Col. 6, lines 3 - 8). Both the task queue of Nelken and the storage media of the application permit the storage of prioritized text and the subsequent selection of text in an order determined by the user, which is not necessarily the order imposed by the priority (see Nelken at Col. 3, lines 43 – 47; Col. 4, lines 8 - 10; Col. 5, lines 1 - 11) (see the application at page 15, lines 20 – 29). Further, the application teaches that the text can be stored in priority order in the storage media (see the application at page 25, lines 8 - 9).

[0007] Regarding the terms "task" and "text," it is my experience that the ordinary meaning of a "task" is an action or a function to be performed or being performed. See Webster's Computer & Internet Dictionary 547 (Random House, 3d ed. 1999); The American Heritage Dictionary 1245 (Houghton Mifflin Company 2d College ed. 1985). The specification of the Nelken patent expressly supports this definition at Col. 1, lines 47 - 48, where a task is defined to include "an action that is to be performed by an agent or an electronic system." However, Nelken has extended this definition by stating that, as far as his system is concerned, a task "may be a piece of data that must be acted on in some fashion" (Nelken at Col. 1, lines 49 - 50). Notwithstanding this, the actual use of the term, "task," within the Nelken patent is limited to text to be prioritized, which is also the subject matter of the application.

[0008] Based on my experience, the systems in both Nelken and the application are operating on electronic documents. The application explicitly makes this clear by saying that text and documents are interchangeable terms (see the application, page 5, lines 18-20). It is well- known in the art that documents may contain both textual data and metadata (additional data about the document). The system in the application operates on textual data (see the application, page 12, lines 17-18) and additional metadata about the text (see the application, page 13, lines 5-25).

[0009] Upon examining Nelken, I find that it only teaches that the system operates on textual data plus possible metadata, not on generic actions or any other type of data. First, let us examine Nelken to determine whether tasks include textual data. At Col. 1, lines 53 - 57, Nelken discloses that the decision engine responsible for receiving and assigning a priority to each task includes a natural language processor for parsing text into concepts and relationships. As is well known in the art of language processing, a natural language processor is structured to process text expressed in a natural language (see Nelken at Col. 1, lines 56 - 57) and is not equipped to process or interpret an action, whether the action is a physical movement by an agent or the abstraction of an action to be performed in the future. The prioritization system of Nelken has a task parser that includes a natural language processor for analyzing the content of text communications expressed in a natural language (Col. 4, lines 24 - 26). It is the task parser that forwards each parsed task to a priority module for comparing the parsed tasks with its priority data and for assigning a priority to each task (Col. 4, lines 38 - 40). The only disclosed path for sending a task to the priority module is from the task parser (Figure 3; Col. 4, lines 38 - 39), and the only disclosed method for parsing each received task into the required concepts, relationships, and keywords is with the natural language processor of the task parser (Figures 3 & 4; Col. 4, lines 19 - 29; Col 5, lines 41 - 49). There is no explanation within Nelken describing how an "action" could be parsed into concepts, relationships, and keywords, unless the action was in the form of textual data.

[0010] Col. 2, lines 55 – 59 also support my understanding that the tasks of the Nelken patent are directed to text-based communications and not actions. Here Nelken notes that communications received by its contact center may be in the form of tasks. Nelken expressly limits these received tasks to text-based communications in the following sentence by stating that "[a]lthough tasks in particular are discussed here, other types of text-based communications, for example remote employees' reports, are within the scope of the invention " (emphasis added). In other words, tasks are treated within the Nelken patent as being a species of text-based communications.

[0011] Nelken goes on to note that its contact center can receive voice communications, but that any such voice communications are converted into text prior to processing by its prioritization system (Col. 2, lines 59 - 63), in order for the voice communication to be parsed into concepts (Col. 4, lines 35-36). Nelken adds that the voice communication can, in addition to being converted into text, have its emotional content determined (Col. 4, lines 32-35). This emotional content is an example of metadata about the textual data.

[0012] Given that the description of Nelken "tasks" is limited to text plus possible metadata, the specification limits the "tasks" of Nelken's claim 1 to be documents, not actions, and therefore equivalent to "text".

[0013] I have been asked to comment on whether there is a distinction between Nelken and the present invention in that "Nelken teaches monitoring how tasks are selected from the 'task queue' whereas the invention teaches watching the user's interaction with text and prioritizing according to the tracking and then feeding that back to the text classifier." (Office Action at page 4). I find that there is no distinction between the monitoring and reprioritizing functions claimed by Nelken and by the present invention. Both systems provide for monitoring text selected by a user from storage (Nelken at Col. 3, lines 10 - 12; Col. 5, lines 1 - 4; Col. 6, lines 11 - 18) (application at page 8, lines 5 - 15; page 15, lines 20 - 27). Both systems then utilize the text selection information to update the guidelines for determining the priority of received text (Nelken at Col. 2, lines 1 - 6; Col. 5, lines 4 - 13; Col. 6, lines 18 - 22) (application at page 15, lines 26 - 30; page 26, lines 11 - 15).

[0014] It is my opinion that claim 1 of the Nelken patent and claim 36 of the application are directed to the same system for automatic prioritizing of text.

[0015] I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Respectfully submitted,

By: John Platt

Date: 20 Jan 2004

Patents Granted

1. 6,674,436 Methods and apparatus for improving the quality of displayed images through the use of display device and display condition information
2. 6,624,828 Method and apparatus for improving the quality of displayed images through the use of user reference information
3. 6,393,145 Methods apparatus and data structures for enhancing the resolution of images to be rendered on patterned display devices
4. 6,380,929 Pen drawing computer input device
5. 6,360,023 Adjusting character dimensions to compensate for low contrast character features
6. 6,327,581 Methods and apparatus for building a support vector machine classifier
7. 6,192,360 Methods and apparatus for classifying text and for building a text classifier
8. 6,161,130 Technique which utilizes a probabilistic classifier to detect "junk" e-mail by automatically updating a training and re-training the classifier based on the updated training set
9. 6,148,104 Incremental ideographic character input method
10. 6,028,959 Incremental ideographic character input method
11. 5,926,566 Incremental ideographic character input method
12. 5,861,583 Object position detector
13. 5,854,625 Force sensing touchpad
14. 5,812,698 Handwriting recognition system and method
15. 5,381,515 Two layer neural network comprised of neurons with improved input range and input offset
16. 5,331,215 Electrically adaptable neural network with post-processing circuitry
17. 5,303,329 Continuous synaptic weight update mechanism
18. 5,204,549 Synaptic element including weight-storage and weight-adjustment circuit
19. 5,165,054 Circuits for linear conversion between currents and voltages
20. 5,126,685 Circuits for linear conversion between voltages and currents
21. 5,107,149 Linear, continuous-time, two quadrant multiplier

16. J. C. Platt. *Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods*. In A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, Advances in Large Margin Classifiers, pages 185--208. MIT Press, 1999.
17. J. C. Platt, *Fast training of support vector machines using sequential minimal optimization*, in Advances in Kernel Methods -- Support Vector Learning (B. Schölkopf, C.J.C. Burges, and A.J. Smola, eds.), MIT Press, 185-208, 1999.
18. S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. *Inductive learning algorithms and representations for text categorization*. In Proceedings of the 7th International Conference on Information and Knowledge Management, 1998.
19. J. Platt, *How to Implement SVMs*, IEEE Intelligent Systems Magazine, Trends and Controversies, Marti Hearst, ed., vol 13, no 4, 1998
20. J. Platt, N. Matic, *A Constructive RBF Network for Writer Adaptation*, Advances in Neural Processing Systems 9, pp. 765-777, 1997.
21. J. Platt, T. Allen, *A Neural Network Classifier for the I1000 OCR chip*, Advances in Neural Information Processing Systems 8, pp. 938-944, 1996.
22. S. Nowlan, J. Platt, *A Convolutional Neural Network Hand Tracker*. In Tesauro, G., Touretzky, D., and Leen, T., editors, Advances in Neural Information Processing Systems 7, pages 901--908, San Mateo, CA. Morgan Kaufmann, 1995.
23. R. Wolf and J. C. Platt. *Postal address block location using a convolutional locator network*. In J. Cowan, G. Tesauro, and J. Alspector, editors, Advances in Neural Information Processing Systems 6, pages 745-752. Morgan Kaufman, San Mateo, CA, 1994.
24. J. Anderson, J. C. Platt, and D. Kirk. *An analog VLSI chip for radial basis functions*. In J. Hanson, J. Cowan, and C. L. Giles, editors, Neural Information Processing Systems, pages 765--772. Morgan-Kaufman, Palo Alto, 1993.
25. J. C. Platt and F. Faggin. *Networks for the separation of sources that are superimposed and delayed*. In J. Moody, S. Hanson, and R. Lippmann, editors, Advances in Neural Information Processing Systems 4. Morgan-Kaufmann, 1992.
26. J. Platt. *A generalization of dynamic constraints*. CVGIP: Graphical Models and Image Processing, 54(6):516--525, 1992.
27. J. Platt. *A resource-allocating network for function interpolation*. Neural Computation, 3:213--255, 1991.
28. J. C. Platt. *Learning by combining memorization and gradient descent*. In R. P. Lippman, J. E. Moody, and D. S. Touretzky, editors, Neural Information Processing 3, pages 714--720. Morgan Kaufmann, San Mateo, CA, 1991.
29. D. Terzopoulos, J. Platt, K. Fleischer, *Heating and Melting Deformable Models (from Goop to Glop)*. In Graphics Interface '89, pp. 219--226, June 1989.
30. J. C. Platt, A. H. Barr. *Constraint methods for flexible models*. Computer Graphics, 22(4):279--288, 1988.
31. D. Terzopoulos, J. Platt, A. Barr, K. Fleischer, *Elastically deformable models*, Computer Graphics, 21(4), July 1987.

Papers Published

1. C.J.C. Burges, J.C. Platt, S. Jana, *Distortion Discriminant Analysis for Audio Fingerprinting*, IEEE Trans. on Speech and Audio Processing, Vol. 11, No. 3, pp. 165-174, 2003.
2. P.Y. Simard, D. Steinkraus, J.C. Platt, *Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis*, Intl. Conf. Document Analysis and Recognition, pp. 958-962, 2003.
3. E. Harrington, R. Herbrich, J. Kivinen, J. Platt, R. Williamson, *Online Bayes Point Machines*, Seventh Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 241-252, (2003).
4. C.J.C. Burges, J.C. Platt, and S. Jana. *Extracting noise-robust features from audio data*. In Proceedings of ICASSP, pp. I1021-I1024, 2002.
5. J. C. Platt, C. Burges, S. Swenson, C. Weare, and A. Zheng. *Learning a Gaussian process prior for automatically generating music playlists*. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, Advances in Neural Information Processing Systems 14. MIT Press, 2002.
6. N.P. Matic, J.C. Platt, T. Wang, *QuickStroke: An Incremental On-line Chinese Handwriting Recognition*, Int'l Conf. On Pattern Recognition, 2002.
7. B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, *Estimating the Support of a High-Dimensional Distribution*, Neural Computation, v 13, no 7, pp. 1443-1472, 2001.
8. H. Attias, L. Deng, A. Acero, and J. C. Platt, *A New Method for Speech Denoising and Robust Speech Recognition using Probabilistic Models for Clean Speech and for Noise*," in Proc. Eurospeech, vol. 3, pp. 1903-1906, 2001.
9. H. Attias, J. C. Platt, A. Acero, and L. Deng. *Speech denoising and dereverberation using probabilistic models*. In Advances in Neural Information Processing Systems 13. 2001.
10. J. C. Platt, *Optimal Filtering for Patterned Displays*, IEEE Signal Processing Letters, vol. 7, no. 7, pp. 179-80, 2000.
11. C. Betrisey, J.F. Blinn, B. Dresevic, B. Hill, G. Hitchcock, B. Keely, D.P. Mitchell, J.C. Platt, T. Whitted, *Displaced Filtering for Patterned Displays*, Proc. Society for Information Display Symposium, pp. 296-299, 2000.
12. J. C. Platt. *AutoAlbum: Clustering digital photographs using probabilistic model merging*. In Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, pages 96--100. IEEE, 2000.
13. B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt. *Support vector method for novelty detection*. In S.A. Solla, T.K. Leen, and K.-R. Müller, editors, Advances in Neural Information Processing Systems 12, pages 582--588. MIT Press, 2000.
14. J. C. Platt, N. Cristianini, and J. Shawe-Taylor. *Large margin dags for multiclass classification*. In Advances in Neural Information Processing Systems 12, S. A. Solla, T. K. Leen, and K.-R. Müller, eds, Cambridge, MA, MIT Press, 2000.
15. J. C. Platt. *Using sparseness and analytic QP to speed training of support vector machines*. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, Advances in Neural Information Processing Systems 11. MIT Press, 1999.

C.V. of John C. Platt

Work Experience

1997-present **Senior Researcher, Microsoft Research**

Invented critical new technologies for incorporation into Microsoft products. Manages a team of 4 researchers and software developers.

Major accomplishments:

- Co-invented ClearType: a method for increasing LCD resolution purely in software
- Invented Sequential Minimal Optimization, a method for speeding up Support Vector Machine learning algorithm by up to a factor of 1000.
- Co-invented RARE: a system for identifying clips in distorted audio streams
- Co-developed first version of Microsoft spam detector
- Invented AutoAlbum, a method for detecting events in digital photographs
- Co-invented AutoDJ, a system for automatically generating music playlists
- Filed 35 patent disclosures

1989-1997 **Director of Research and Principal Scientist, Synaptics, Inc.**

Managed 8-person research department. Member of Management team of company. Set technological direction for company.

Major accomplishments of team:

- Developed QuickStroke, world's first Chinese handwriting recognition system that anticipates what character user is writing
- Developed Xpad, a force sensor that is compatible with Synaptics capacitive touchpad chip
- Created multiple custom neural network ASICs for customers
- Wrote internal software tools for entire company

Summer 1988 **Consultant, Apple Computer**

- Developed physically-based computer graphics models on Cray

Summers 1985-6 **Intern, Schlumberger Artificial Intelligence Lab**

- Co-invented elasticity-based computer graphics models

Educational Experience

- Ph.D. in Computer Science, Caltech, 1989
  - Thesis title: Constraint methods in computer graphics and neural networks
- M.S. in Computer Science, Caltech, 1985
- B.Sc. in Chemistry, California State University Long Beach, 1982
  - Outstanding graduate, School of Natural Science
  - Graduated at age 18

Academic Participation

- Co-chair of Neural Information Processing Systems conference, 1991, 1994, 2002.
- Action Editor, Neural Computation journal
- Member, Applied Vision technical committee, Society of Information Display
- Member of program committee for UAI, WWW, ICML and other conferences.